January 1991

DCIEM No. 91-05

AD-A237 363

DTIC
ELECTE
JUN 2 6 1991
S C D

Shape Recognition by Computer in Simulated Aerial Images

T.R.H. Cutmore

Accession For
NTIS GRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification

By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

Defence and Civil Institute of Environmental Medicine
1133 Sheppard Avenue West, P.O. Box 2000
North York, Ontario M3M 3B9

91-03487

DEPARTMENT OF NATIONAL DEFENCE - CANADA

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Table of Contents

Abstract i

1. Introduction 1
1.1 Model-based Vision 1
1.2 Levels of Resolution 2

2. A 2-D Solution for Constraining Component Relations 3
2.1 Extending RBC Theory 3
2.2 Basic Definitions 3
2.3 Feature Space 5
2.4 Algorithm 6
2.4.1 Data Structures for Models and Constructions 7
2.4.2 Using Shape Information 7
2.4.3 Image Interpretation 8

3. An Application 9
3.1 Input 9
3.2 Geon Types and Instantiations 9
3.3 Models and Constructions 9
3.4 Relations 10
3.5 Shape Parameterization and Feature Space 10
3.6 Some Example Images for Recognition 11

4. Conclusions 14

References 28

Abstract

Visual recognition of objects by a machine involves classifying an input using knowledge about the kinds of objects expected in the domain. Model-based systems maintain a knowledge of objects in the domain in the form of a representation which can be compared to the unknown input. Since a given object type may appear in a variety of forms and under a variety of viewing conditions some efficient yet flexible means of guiding the recognition process to consider and then verify the object identity is necessary.

The utility of low-resolution shape information to constrain object recognition was investigated in the context of a system which is predicated upon a component description of objects. A computationally intensive prepass using a syntax for combining components yields a universe of "constructions" which are coded into a construction relation feature (CRF) map. Each construction is coded into the N-dimensional map according to a shape parameterization of its low-resolution image (each dimension codes a shape feature). From a subset of these constructions the object models are specified in terms of their component structure. The CRF map thus links the low resolution shapes of instances of an object to its object model.

To recognize an unknown object the input is first converted to low resolution. Then, shape parameters are taken (for example, in terms of its relative elongation and compactness). The CRF map is examined in the region of these feature coordinates for possible matches with object models, first on the basis of a component match followed by a verification of the relations between components. An application to identifying objects in simulated aerial photographs of military airports is presented.

1. Introduction

1.1 Model-based Vision

Computational systems which emphasize knowledge about *objects*[1] in the world as important in the process of image *interpretation* (or object recognition) are regarded as "model-based". This influential approach in computational vision research began in 1965 with Roberts and has continued to receive much attention (e.g., Kanade, 1977; Brooks, 1981; 1984). More recently, Biederman (e.g., 1985; 1987) has been developing a psychological theory of model-based object recognition predicated upon *component* analysis (RBC theory). His work has examined the utility of breaking the recognition process into the stages of: segmenting the *object* into 3-dimensional primitives called geons[2], determining their general spatial *relations* to each other, then matching the resultant to models of *objects* in the world. Browse and colleagues have been investigating the computational virtues of this approach along with the usefulness of considering multiple *resolution* image analysis (Smith & Browse, 1988; Browse and Rodrigues, 1987; Browse, 1982). One of the major difficulties uncovered in this analysis of RBC theory is its difficulty in effectively handling the important problem of *component relations*. Although Biederman made suggestions as to how these *relations* might be computed, it is apparent that the suggested solutions detract from the simple elegance of the main theory. The problem may be stated thus: In order to unambiguously recognize an *object* using a list of its *components* it is necessary to also incorporate information about their spatial organization if the *object* is to be discriminated from others with the same *components*. This is particularly an issue in RBC theory where the geons that compose the *objects* have very general descriptions making it likely that in any domain of moderate complexity there will be many *objects* consisting of the same *components*.

Biederman suggested using "general relations". *Component* spatial *relations* would be described in terms of being "near" to another, or connected by the "end", or "middle" etc.; where the concepts in quotes do not have a specific numerical value but, in some fashion, encode the approximate relative position of *components* in an *object*. The problem in attempting this, however, is in deriving the general relations initially. It is not apparent how this may be done other than by first computing an exact measure and then categorizing this using a threshold technique (for example to decide if a *component* connects by its "end" or its "near-end" or its "middle" etc.). Exact measures are computationally expensive however, (see Marr & Nishihara, 1978) and thus detract from the elegance of the theory and would clearly slow the speed of recognition.

The RBC approach has computational appeal if an efficient means can be found to constrain *component rela-*

1. An appendix is provided defining common names (italicized) which have a specfic technical meaning in this document.
2. A term coined by Beiderman. In this document it refers to a simple two-dimensional geometric shape such as a circle, square, triangle etc. which can be used in combination to describe more complex shapes.

tions. This report describes a solution which utilizes low *resolution* images of an *object* as an estimate of its general shape and thus its *component relations*. To facilitate study of this method a two-dimensional (2-D) domain was selected, thus obviating the complexity of multiple views of an *object* in a 3-D domain without violating the spirit of basic principles. A possible 2-D application area of military interest was examined (section 3) with a prototype system.

1.2 Levels of Resolution

Given that a low *resolution* image is desirable for the purpose of constraining *component relations*, the question arises: How low *resolution* should such an image be? This question may be addressed by considering a possible hierarchy of *component resolutions*. The lowest possible *resolution* is one that permits no discrimination among shapes for a given method of analysis (fig. 1). The next level of *resolution* would permit a few shapes to be discriminated. For example, an ellipse, triangle and rectangle might be selected on the basis of it being possible to define these as having two, three and four "vertices" respectively[1]. It will be convenient to refer to this as the "type" level. A simple description of a wide range of *objects* would be possible if the size of these rough *components* were allowed to vary in coarse steps. At the next higher level of *resolution* each of these *component types* could be resolved into variations, for example, in terms of the ratio of major and minor axis for ellipses. This will be referred to as the "instantiation" level. Again, variation in scale for each instance would allow a variety of *components* to be defined. Further levels could be defined similarly, defining ever more precise *component* descriptions. However, the instance level may be argued to provide the appropriate level of *resolution* for the purpose of an RBC system since the variety of *components* should be moderate, perhaps ten to twenty in number (Biederman, 1987).

In correspondence with the above reasoning, a low *resolution* image of an entire *object* may only be interpreted at a very general level (e.g., an "aircraft") and high *resolution* images may be interpreted as specialized versions of the *object* (e.g., "F-18" vs. "Cessna 180"). This principle has been utilized since Kelly (1971) and Tanimoto and Pavlidis (1975), but has been made most explicit in the work of Browse (1982), Neveu et al., (1986) and Browse and Rodrigues (1987). Thus, low *resolution* shape information could be used to constrain the class an *object* belongs to but might not be expected to resolve specific members of the class.

In summary, RBC appears to provide a useful framework for developing a computer-based recognition system. However the following difficult and unsolved problem first requires a solution: What kind of information can be economically extracted from a coarse level image which will naturally constrain *component relations* at higher *resolutions*? Without a systematic handling of the relational aspects of *object* models there could only be unambiguous recognition of those simple *objects* which require only a single geon for their description. It has been argued that Biederman's concept of "general relations" such as "near to" would still involve extensive computation. It is apparent that an appropriate low *resolution* image of an *object* may constrain the *relations* of its components. In the next section the

1. In the case of an ellipse the vertices might be defined to be the end points of the major axis.

development of an algorithm to utilize this low *resolution* information is described. The effectiveness of using low *resolution* shape information in the context of an RBC theory is tested in section 3.

2. A 2-D Solution for Constraining Component Relations

This section describes, in general terms, an algorithm for using low *resolution* shape information to help *interpret* the identity of an unknown *object*. This paper is based on Cutmore (1989) and a formal computational description is in preparation by Browse and Cutmore. Biederman's RBC system did not include the concept of multiple *resolutions*, thus the system described below is referred to as an extended RBC system. To expedite the presentation definitions of the major system elements are provided. Feature space is introduced since it forms the basis for the primary data structures. This is followed by the solution algorithm.

2.1 Extending RBC Theory

An idea of the overall system within which a relation-constraining subsytem could contribute is as follows: The method of extension of RBC theory to include multiple *resolution* is relatively straightforward. A separate system which determines the identity of the components of an unknown imaged object in high *resolution* (but not their spatial relations) could operate in parallel with the system to be described in this report (see Smith and Browse, 1987). This system may be referred to as an RBC-C system. A smaller set of "low-*resolution*" geons could then be defined, and simpler image features could be utilized to determine their identity in the same way the fine-level geons are detected, plus a method which constrains their spatial relations. A mapping would then be required between these lower *resolution* geons and the high *resolution* geons identified by RBC-C (see, Browse, 1982). The RBC system which explicitly includes a method for constraining component *relations* using low *resolution* shape will be referred to as RBC-R. It was also of interest to determine to what degree, simply considering low *resolution* shape constraints could provide an interpretation. The RBC-R system is described below and tested in section 3 in isolation of the other programs with which it could be coordinated.

2.2 Basic Definitions

Five kinds of entities will be referred to: geon *types*, geon *instantiations* (or *components*), *component relations*, *constructions* and *object* models. Geon *types* as discussed above are primitives analogous to Biederman's RBC theory. The set of geon *types* G is defined as:

$$G = \{g_1, g_2, \ldots, g_i, \ldots, g_m\} \qquad (1)$$

From arguments presented above, a *component* is an *instantiation* of some geon *type*. The set of components is defined as:

$$C = \{c_{11}, c_{12}, ..., c_{21}, c_{22}, ..., c_{ij}, .. ,c_{mn}\} \quad (2a)$$

Thus, each g_i is a geon *type* under which are nested a set of c_{ij}. For example, g_1 may be an ellipse *type*, and c_{1j} various specific ellipses. Since a *component* may appear more than once in a list some means of distinguishing them is necessary. The actual use of a set of *components* to specify an *object* is referred to as a c-list:

$$\text{c-list} = \langle c_1, c_2, .. , c_i, .. , c_s \rangle \quad (2b)$$

Thus, replications of a c_{ij} are permitted but distinguished and s is the number of *components* in an *object*. A c-list is distinguished from C by having only a single subscript for its members.

A spatial *relation* between *components* is designated by: $R(c_i,c_j)$; one of which is considered fixed with the other's position defined relative to it. Two-dimensional *objects* defined in a Cartesian plane consist of 2-D *components* which bear spatial *relations* defined in terms of three parameters which describe the relative displacement of a "movable" *component* relative to a fixed one. Each *component* has a coordinate frame and the relative location of *components* are represented as the transformations necessary to take one *component* coordinate frame into another. The parameters in 2-D are: displacement in x (Δx), displacement in y (Δy) and rotation about some point in the plane, $\Delta\theta$. In the methods described below, this rotation will be about a "connection point" between two components[1]. The displacements are defined relative to standard initial positions of the components. For example, the initial position of the two *components* may have their centroids at a common point. A *relation* between *components* c_i and c_j is given by $R(c_i,c_j)$, which consists in a triplet of two translation operations and a rotation. A *construction* is defined in terms of a list of *relations* between *component* pairs:

$$K_i = \langle R(c_a,c_b), .. , R(c_i,c_j), .. \rangle \quad (3)$$

A *construction* may be conveniently described when successive elements are appended one at a time to the list much as a real *object* might be assembled, (fig. 2). Connection points can be specified on each *component* as offsets with respect to the centroid. This ensures that *relations* between *components* are restricted such that all *components* will be a part of a single whole. Rotation of one *component* (designated the movable component) occurs about the coincident connection points of two components. For K_i with n *components*, n-1 *relations* are required to define the *construction*.

1. The use of defined connection points ensures that a construction will be a connected whole. Thus, a single silhouette shape will always be produced.

A K_i is thus a set of connected *components* with the spatial *relations* between them specified as in eqn. 3. An *object* model is defined by a non-empty list of *constructions* with replications permitted:

$$M_r = \langle K_a, K_b, .. K_i, .. K_s \rangle \qquad (4)$$

A set of *object* models can be specified for an application domain of RBC-R. This would be in the form of:

$$M = \{M_1, M_2, .., M_r, .. M_n\} \qquad (5)$$

where M_i are defined as in eqn. 4, plus a label for each. As a matter of convenience, an *object* model may be represented as a list of *components* along with ranges of articulation of *components* defined for each list. More than one list may be necessary since some models may have instances composed of different lists of components.

2.3 Feature Space

The utility of feature space for pattern classification has been investigated (see Horn 1986; Ballard and Brown 1984 for overviews of basic methods). In brief, this approach takes a set of N measurements on a sample from the domain of patterns. In visual domains these measurements may be shape measures, number of vertices, number of concavities etc. An N-dimensional feature space may then be used to code a pattern as a vector in that space. Typically some form of cluster analysis is then done to identify boundaries between sets of points in the space which allows discrimination of the input patterns into their appropriate categories. To classify (or recognize) an unknown input, its membership to a region in feature space is computed. This technique will work well if one can find features which result in boundaries in feature space. However, this is no simple task and the types of features that prove useful in one domain may be inappropriate in another. However, it may be possible to identify (in the present case) shape features that are useful in partially classifying or discriminating between images of different types of *objects*.

The question of interest here is: "What kinds of shape measures would be useful in coding gross spatial *relation* properties of components?" Some considerations are (a) the shape parameters need not extract information such that the original image could be reconstructed such as are described in Ballard and Brown (1982). Partial information about the shape will suffice (b) It would be desirable for the shape measures to be invariant over the affine transformations of: translation, rotation and scale. This would allow the imaged *object* to be less constrained in its presentation. (c) Finally, the constraint of no object occlusion may be assumed[1].

1. Occlusion is a technical term in computer vision. It refers to the overlap in the images of objects in a scene. The nearer object is said to occlude the more distant one. If a 2-D domain is being analyzed then this is a reasonable assumption.

Ballard and Brown list a variety of shape measures which satisfy the above conditions. In addition, other recent work has extended this range (e.g., Goshtasby, 1985; Bhanu 1984; Bhanu & Faugeras (1984); Tsai and Yu 1985). Wang, Magee and Aggarwal (1984) describe a method which illustrates the value of taking shape measures on 2-D silhouette images. This is of interest since silhouettes would be expected to be sensitive to *component relations* and offers a simple image type for processing and may thereby improve processing speed.

Ma, Wu & Lu (1986) summarize the following desirable properties of shape descriptors:

(1) The features should be independent of translation, rotation, scale and cyclic shift of the starting point for computation.

(2) The distinction between the features of two *objects* which have different shapes should be as large as possible.

(3) The number of features used in classification and recognition should be as small as possible.

(4) Computational time and storage capacity required are short and small respectively.

To this list it would also seem important to add that the shape features should tap essentially independent kinds of information, unless redundancy is explicitly desirable (which may be the case in inherently "noisy" domains). Two shape descriptors which meet these criteria are presented in section 3.

2.4 Algorithm

The following algorithm could be used in isolation for providing coarse discriminations and *object* labels, such as: airplane vs. cruise missile vs. tank. It is properly a subsystem of a complete extended RBC system. Biederman's RBC theory describes how to identify the geons individually. For example, from an image of an airplane, two rectangular geons (the main and tail wings) and an ellipse (the fuselage) may be identified. The RBC-R algorithm does not include this capability; it is described elsewhere (Smith & Browse, 1988). The RBC-R algorithm could work in concert with such a geon identifying program, but it need not. Together, the two systems would embody the main principles of the extended RBC theory.

An introductory overview of RBC-R is as follows: In the first phase a data structure is created. A universe of all legal combinations of geon *instantiations* is constructed. This is the set of all possible K_i which could be defined for some domain of interest. Many K_i will belong to an initial set of *object* models as in eqn. 4. Many may belong to no model initially, but would provide elements for new models which could be added to the domain at some later point in time. This universe of K_i is then systematically organized into a feature space. This is accomplished by extracting features from each K_i, using these as coordinates into the feature space, and locating at this point the *components* and *relations* of the K_i. In this way a data structure for encoding K_i configurations is created. This structure is computed once for the domain. In the *object* recognition phase, an image of an unknown *object* is presented. It is converted to a lower *resolution* and features extracted. This provides an index into the feature space to retrieve a set of K_i. One of these K_i should be an element of an *object* model which is the correct *interpretation* of the image. Methods for resolv-

ing *component* and *relation* consistency are applied to this set in order to achieve the *interpretation*.

2.4.1 Data Structures for Models and Constructions: The definition of most *objects* includes some variation in the *relations* among components. For example, the rotor of a helicopter may rotate through 2π. For M_r, this variation is described in terms of an extended list of K_i. Since the low *resolution* analysis will not distinguish fine variations in *component relations*, model *relations* need not be specified in fine detail and thus, the M_i list may be kept to a manageable length. A data representation for the description of M_i *component relations* is a set of n-matrices (Kron, 1939) where n is the number of degrees of freedom of *component relations* (fig 2). The cardinality of this set equals the number of distinct *component* sets for M_i. A 2-D *object* with two *components* will have three degrees of freedom (two translational, one rotational) for its various configurations. If a third *component* is added, three more degrees of freedom are needed to describe the *relations* of this *component* relative to one of the other two. For example, a 3-*component* K_i could be specified in terms of any two of $R(c_1,c_2)$, $R(c_2,c_3)$ and $R(c_1,c_3)$. A six-matrix would be required to store the relational quantities. If a different set of *components* were also needed for that same M_i a second n-matrix would be needed for storing these *component relations*; thus the cardinality of the n-matrix set would be two.

Each dimension of an n-matrix has a metric defined on it appropriate to the relational quantity which that dimension encodes. In the case of translations this could be coordinates of connection points (relative to the *component* reference frame) and for rotation, increments of angle. If a relatively coarse description of the parameter space is assumed, then the n-matrix will have discrete dimensions each with a reasonably small number of levels. For example, the rotor connections in the "x" direction with respect to the fuselage in fig. 3, could be defined to have three connection points. Thus, the dimension of the n-matrix which encodes this parameter would require only three levels.

To summarize, the foregoing has described how 2-D *object components* and *relations* may be represented. A particular K_i is encoded as an element of an n-matrix. *Relations* between a particular set of *components* are encoded as a single n-matrix and if more than one *component* set is required, more n-matrices are used. What next needs to be determined is a method by which RBC-R model knowledge may be accessed using information derived from analysis of the low *resolution* shape of an *object*. It is also desirable that the capability to add new models to RBC-R be considered in the solution.

2.4.2 Using Shape Information: To satisfy these requirements, an exhaustive prepass is performed which analyzes the full range of *component* configurations. This universe of K_i is defined by specifying the set of *component instantiations* from a set of *types*. These two levels of the *component* hierarchy need to be carefully considered for the domain of application. The main criterion is that the *component* set be rich enough to model all known *objects* of interest within the domain and furthermore, anticipate the addition of new models. The manipulations of these *components* to form K_i is described by a *component* relational grammar which generates the full set of "legal" *component* configurations for the domain. Each K_i so created yields an image which is converted to a lower *resolution* and is analyzed in the form

of a single silhouette with Euler number[1] one. n shape measures are applied to acquire an n-tuple coordinate into an n-dimensional feature space. This space is quantized to a degree appropriate for the discrimination task at hand. This quantization may best be determined by considering the range of shapes to be discriminated relative to the fineness of discrimination. The minimum and maximum shape parameters computed for a given domain could be used to define the the limits of the space, and the number of resolvable levels on any dimension could be decided empirically in a calibration of the system. This approach was followed in the test environment in section 3.

The K_i *components* and *relations* which yielded these features are stored at the feature-indexed n-tuple coordinate of the feature space. The resulting structure is referred to as a *construction-relation* feature *map* (CRF *map*). This process will be computationally intensive, even for a relatively simple domain since the variety of *component* combinations and configurations grows geometrically with number of *components* and permissible *relations*. However, it need only be computed once and thereafter remain a static structure to be used in the process of *object* recognition.

2.4.3 Image Interpretation: After creation of the static CRF *map* an unknown *object is* presented to RBC-R in the form of a silhouette image. The major steps in the recognition phase are:

1. An input image is provided in a form which allows for conversion to a lower level of *resolution*.

2. Shape parameters are extracted from this second image and used to index the feature space.

3. The feature space n-tuple is examined for an associated set of *construction* (K_i) lists. If this list is non-empty then,

4. Each K_i's c-list is compared against the *object* model c-lists to determine whether models have been defined which have the same components. If this step produces an intersection list which is non-empty then,

5. The relational n-matrices of each K_i surviving the previous step is compared against the c-list matched models. If a non-empty intersection for one or more of K_i is found then,

6. The process stops and yields models which have satisfied the two necessary criteria of *component* matching with input elicited K_i and *component* spatial *relation* similarity.

If any of the steps (3-5) fail to maintain a non-empty set of *interpretations* then RBC-R can search the local region of feature space (go back to step 2 and choose a nearby N-tuple) until some *interpretation* is resolved or a threshold is passed which indicates that RBC-R is unable to make an *interpretation*.

The reason why this technique works in permitting partial consideration of relational information without actual analysis of the metrics involved in the *relations*, is that the coarse level shape information encodes not only information about the finer-level elements but it also encodes information about their *relations*.

---

1. A technical term in computer science defined as: the difference between the number of distinct shapes in an image and the total number of holes in these shapes.

3. An Application

3.1 Input

The domain for the application was aerial "photographs" of military airport scenes. This domain is suitable for 2-D analysis and provides a diverse set of *objects* for discrimination. The input images processed by RBC-R were not, in fact, real photographs, but rather, bit arrays which encode silhouette images of *objects* with no occlusions. All images were, therefore, simulated and the names given to *object* models are for illustrative purposes only. Figure 4 shows a low-*resolution* silhouette of an "airplane". Conversion of a high-*resolution* image to low-*resolution* was simulated by making relatively coarse images to begin with and smoothing the perimeter with a simple mask. Thus, in what follows the input image for recognition is assumed to be low *resolution*. The K_i were constructed to yield the same type of image for shape analysis during preprocessing. The implementation was written in Common Lisp (Steele, 1984).

3.2 Geon Types and Instantiations

The set of high *resolution* geon *types* used in generating the K_i were: an ellipse, a rectangle and a triangle. These were selected since they can be used to produce a range of different *instantiations* to yield a variety of *objects* that have the appearance of the kinds of things expected in an air field. For example, in fig. 4 the airplane fuselage is an ellipse, the main wings are a single rectangle as are the tail wings. All *objects* and *constructions* were defined in terms of c-lists of *instantiations* of these three geon *types* and each was configured from three components, (fig. 5).

3.3 Models and Constructions

The models in this implementation consisted of a small subset of the total *construction* set. There were, therefore, many K_i which belonged to no model. For illustrative purposes, some models were deliberately constructed out of the same *components* but with different spatial *relations*, for example, the "airport" and the "airplane" models. One model (the "tank") also consisted of two separate lists of components. Four models, "airport", "tank", "fuel truck" and "helicopter" had articulating components. A model was stored in RBC-R as a list of: *components*, *relations* and name.

The *construction* universe was created using a program to draw and transform polygons to form composite silhouette shapes. This was followed by the application of a low pass filter to "smooth" the perimeter of the shape to simulate a low *resolution* image.

In creating the *construction* universe two possible extremes are (a) to create only model *constructions*, in which case a pre-pass will be needed each time a new model is added or redefined or an existing model extended; or (b) the opposite extreme is to compute all possible combinations and *relations* of *components* in the set (within some *resolution*), in which case adding new models would not require another pre-pass. Given that extreme (a) violates the spirit of RBC as a concept of a general and flexible *object* recognition scheme and that extreme (b) would create a vast data base, an intermediate strategy was adopted. A subset of the full range of possible K_i were used. Figure 6 shows the domain hierarchy and the set of nine models used in this implementation. The models were composed from 48 mem-

bers out of a total set of 323 K_i in the *construction* universe which was derived from eight c-lists.

3.4 Relations

The relational n-matrix explicitly encodes in separate dimensions each degree of freedom of *component relations*. For simplicity of construction, one *component* was always the reference with the other two specified in relation to it. In other words, one *component* was held fixed (c1) and the other two (c2 and c3) attached to it. A 6-matrix was therefore required to encode the *relations* of these K_i: two translational and one rotational degrees of freedom for each of the two pairs of components.

The 6-matrix was implemented as a bit array and a *relation* was stored by setting the appropriate bits in this data structure. The size of the dimensions permitted up to three connection points per *component* and ten levels of rotation. The size of the *relation* space is therefore the product of the dimensions: $3^4 * 10^2$ (8100 bits), or approximately 1 kilobyte. This results from specifying 3 connection points on 4 displacement degrees of freedom and 10 angles of rotation on two rotational degrees of freedom. The *relation* space did not contain any information about the identity of the connected components, this was specified separately.

3.5 Shape Parameterization and Feature Space

Two shape features were selected on the basis of considerations outlined in section 2.3. "Compactness" may be roughly defined as the relative efficiency with which a perimeter bounds a closed 2-D figure. In the present context this measure is defined as:

$$Cp = \frac{A}{P^2} \tag{6a}$$

where A is the area of a silhouette image and P is the length of the perimeter. Cp is minimal for a circle ($1/4\pi$). The computation of this measure is straightforward from a silhouette image. The boundary is traced in either clockwise or counterclockwise direction to measure the perimeter and the area can be measured in number of pixels. A normalized measure can be computed as:

$$Cp_n = \frac{A}{P^2} * 4\Pi \qquad (6b)$$

The normalized compactness is derived by multiplying by the inverse of the compactness of a circle.

"Aspect ratio" defines the relative elongation of a shape. A simplistic approach is to consider the ratio of the sides of a rectangle fitted around the perimeter of the shape such that the opposite sides of the rectangle intersect the most distant points of the image. The steps in the computation are thus: Find the points on the perimeter most distant (length = R_l) from each other and construct a line through them. Then, compute the minima and maxima (in y) of the perimeter points (difference = R_w) in the image after it has been rotated and translated to place the constructed line on the x-axis. The normalized aspect-ratio is:

$$Ar_n = \frac{R_w}{R_l} \qquad R_w <= R_l \qquad (7)$$

Eqn. 7 has a maximum value of 1 when the sides are equal. A more precise measure of aspect ratio based on moments is given in Ballard and Brown (1982).

These two shape measures were computed for each K_i after it was constructed and converted to a lower *resolution*. This pre-pass data was stored in a file, recording components, connection points and angles.

The feature space (CRF) *map* was computed by reading the data file and determining the maximum and minimum for each feature. These were then used as the extreme quantized values with all other feature values scaled accordingly. The dimensions of the space was 10 X 10. This relatively conservative size for the CRF *map* was found to be satisfactory for discriminating the shapes in this small domain. If the results of the (computationally intensive) shape analysis of K_i in the prepass are stored in intermediate form, then calibration of the coarseness of the CRF *map* may be done separately. An overview of the steps involved in the recognition of an "airplane" is illustrated in fig. 6. The high *resolution* "photograph" is converted to a lower *resolution* and features extracted which index the CRF *map*. This produces a set of K_i which are then compared against model representations for *component* and *relation* compatibility.

3.6 Some Example Images for Recognition

With the CRF *map* loaded and the model-list defined, test images were given to RBC-R for recognition. These were in the same format as those previously analyzed during creation of the K_i (i.e., bit images of low *resolution* silhouettes).

Fig. 7 (a) shows an image of a "jet" which was presented. This image was constructed using the same program functions as used in creating the K_i universe. Several points are of interest in this test: (1) The correct model is found (2) The *interpretation* is partially ambiguous in the sense that the "airplane" model was also found. This is interesting since these two *objects* intuitively have similar shapes. (3) The *component* list elicited by the image also matched the "airport" model, but was rejected on the basis of non-matching *component relations*. (4) Since *component* matching is performed at the level of geon *instantiations* as opposed to the basic geon *types*, RBC-R does not evoke a match with the "rocket" or "cruise missile" models which share the same geon *type* and relational configuration as the "jet" model. (5) More importantly, the models for the "fuel tanker", "helicopter" and "conveyor" are not evoked as they would be if geon *types* were used as *components* in RBC-R. To see how this would occur, first note that the "airplane" K_i maps to the same point in the CRF *map* as the "jet", (this is why it's model is considered during the recognition process). If geon *types* were *components* the *component* list would evoke the three other models since each is composed of an ellipse and two rectangles, just like the "airplane". These models would be considered as candidates for which the *relation* space should be checked. Although subsequent rejection would occur on the basis of a failure of *relation* match, this extra computation was avoided by using geon *instantiations* instead of geon *types*. Points (4) and (5) underscore the importance of choosing the appropriate *component resolution* and illustrate the power of using the shape information inherent in the different geon *instantiations*. If geon *types* were used it would impossible to discriminate the "fuel tanker", "helicopter" and "conveyor" since these models were each composed of the same geon *types* and *relations*.

Images of a "rocket" and "cruise missile" were presented [fig. 7 (b), (c)] and were correctly discriminated from each other. Each mapped to a different point in feature space. For the rocket image RBC-R rejected "airplane" and "airport" models on the basis of a failure of *relation* matching. In the latter case, a K_i consisting of the list <LONG-ELL LONG-RECT SHORT-RECT> mapped to the same feature coordinates as the "cruise missile". This illustrates RBC-R's capability of correctly rejecting non-*object constructions* during recognition, an important capability if such a system is to operate in a general context where much of the CRF *map* contains such *constructions*.

Scale and rotation invariance is illustrated in the next set of examples. Fig. 8 shows the results of testing "airplane" images at various scales and rotations relative to the K_i which was originally used to create the airplane shape for the CRF *map*. In other words, these images were never shown to RBC-R during the pre-pass. These tests, therefore, differ from the previous ones in which a sample image from the K_i set was submitted for recognition. As the figure illustrates, the system recognized the scaled-only images after a single n-tuple index to the CRF *map*. Consistent with

the testing of the "jet" image noted above, both the "airplane" and "jet" models were evoked. This was because the K_i shape features for the plane and jet both mapped to the same place in feature space. The "airport" model was evoked because the *component* list of: <LONG-ELL LONG-RECT SHORT-RECT> has a K_i at the computed coordinates and all models with these *components* were examined. It was rejected on the basis of a failure to match *relations*.

Of greater interest is the test result from the scaled and rotated image. This test result is typical of many others in which combinations of scale and rotation transforms were used. Appendix 2 contains an actual run on this image. The initial feature coordinates were not directly on target to find the "airplane" model. RBC-R searched the surrounding CRF *map* region by way of off-set values from the initial coordinates. To be fair, all eight surrounding points were examined in the *map* (since the starting point is arbitrary). The models evoked were the expected ones: "airplane" and "jet".

The "tank" model was defined here to suggest how this 2-D system might be extended to a 3-D world. Fig. 5 shows the tank model composed of two c-lists: each has the same body and turret *components* but differ in the gun *component* (two gun elevations were used: 0 degrees and 45 degrees). Two-D images of 3-D scenes contain information about the 3-D *components* through projective transformations. Of course, some information will usually be lost in such a transform. However, there may be ways of making use of what information is retained using knowledge of projective geometry. In this simple example the change in 2-D *component* length was used encode a change in 3-D *component* orientation. In general, an RBC-R system could store a list of shapes that would be expected in images of 3-D *objects* in some domain. If a low *resolution* image filter were used this would limit the range of the images that would need to be stored and thus may be a practical extension of the, essentially, 2-D system described here to 3-D domains.

The complete set of K_i used to define the "tank" model are shown in terms of the n-tuples of the CRF *map* in fig. 9, with the two "gun" angles and ten "turret" angles. This figure shows that the CRF *map* (at this quantization of the features) had a good separation of the K_i and these results show the smooth albeit non-linear changes in shape parameters. During recognition RBC-R can not only identify that the image is of a "tank", but also the orientation of the "turret" and "gun" with respect to the "tank body".

It was also of interest to submit an image with an intermediate "gun elevation" of 25 degrees. In this test the feature coordinates mapped to a point between those for the 0 and 45 degree "gun elevations" (fig. 9, bolded square). RBC-R could find no model at the initially computed feature coordinates but following a local search of the CRF *map*, the system finds both the 0 degree and 45 degree "gun elevation" models: an illustration of interpolation.

Finally, to examine the hypothesis that the two shape measures are essentially independent, the correlation between the two parameters was computed. The data was obtained from the pre-pass analysis of the 323 members of the K_i universe. The correlation was found to be 0.26. This indicates that only about 7% of the variance in the two shape

measures is shared and would appear to justify the initial assumption of independence.

4. Conclusions

A few straightforward conclusions ensue from these investigations.

(1) The use of shape information severely constrains *interpretation* of the way that a set of *components* may be assembled. Even small variations in the spatial *relations* between a set of *components* can produce sets of feature co-ordinates that discriminate them, see for example fig. 9.

(2) How these discriminations are used is another issue. Depending on the definition of *object* models the discriminated feature sets may be integrated into a single model, as they were for the "tank" model, for example; or, conversely the sets may be segmented to match different models, as in the case of the "cruise missile" and "rocket".

(3) The implemented system revealed the robustness of these shape-based methods in dealing with new images consisting of scaled and rotated transformations of images of *objects*, (the "airplane" examples). In addition, the system was found to provide reasonable interpolation behavior (due to smooth parameter variations).

(4) The system produced ambiguous *interpretations* for cases that might seem to be intuitively difficult to discriminate (for example, a "jet" verses an "airplane").

(5) Some parameters for the implementation were chosen intuitively with little investigation as to their general applicability. However, some preliminary conclusions can be suggested: (a) the two selected shape parameters appear to be independent, (b) the relatively coarse images (approx. 50 pixels square) appear adequate in terms of the shape information that they contain, (c) a relatively coarse feature space quantization (10 x 10) appeared adequate to permit good discrimination (see figure 9). However, in a domain with many more models, a finer quantization may be necessary.

Future research could investigate how to optimize the recognition process by a systematic study of the effects of changing a number of variables, which in the present case were instantiated intuitively. For example, the effects of altering the quantization of the relational n-matrix and CRF *map* would likely reveal more efficient use of these spaces. The features selected for illustrative purposes here (compactness and aspect ratio) are only two among a range of possible alternatives. Finally, as suggested in the previous section, RBC-R appears amenable to extension to a 3-D world by extending the K_i which define models.

Research on this recognition system is continuing to extend its application to actual photographs. As noted at the beginning of section 3, RBC-R is a system which operates on simulated images. In actual photographs, segmentation of the image into separate regions which might contain an *object* is necessary. There are many techniques already available to achieve this end. A more difficult problem is that real photographs in 3-D domains are, of course, suscep-

tible to all the imperfections inherent in the imaging process, such as, shadowing and other effects of lighting, occlusion of *objects* by one another and incomplete silhouettes of the *objects*. Methods will need to be developed to permit RBC-R to deal with these "noisy" imaging problems. Since RBC-R is designed to operate on low *resolution* images it should be relatively robust, especially in response to noise effects which do not grossly affect overall silhouette shape. Finally, there are many aspects of this system which could be described with a parallel algorithm and further research to study the computational advantages of this should lead to significantly faster recognition.

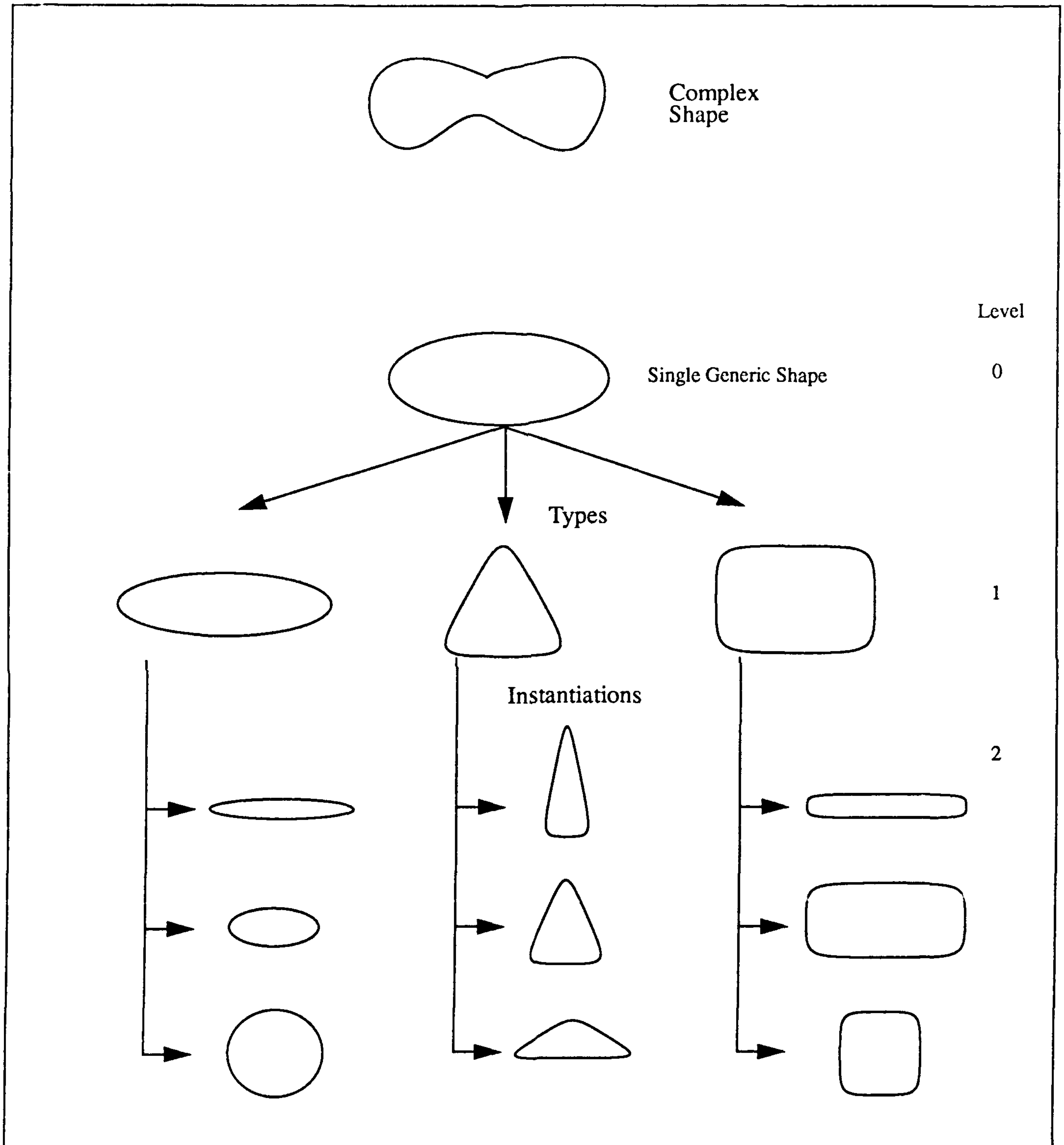

Figure 1: How to represent an arbitrary shape. A Hierarchy of *resolutions* of geons may be considered. At the lowest level of *resolution* (level 0) no discriminations between primitive shapes are possible. At level 1, basic shapes (*types*) may be distinguished. At higher levels, these primitive shapes resolve into finer sets of shapes (*instantiations*) which may be discriminated from one another. In a shape recognition system, a decision must be made as to what level of *resolution* is desired for adequate or optimal system performance. A complex shape as shown above will generally require two or more geons for an adequate representation.

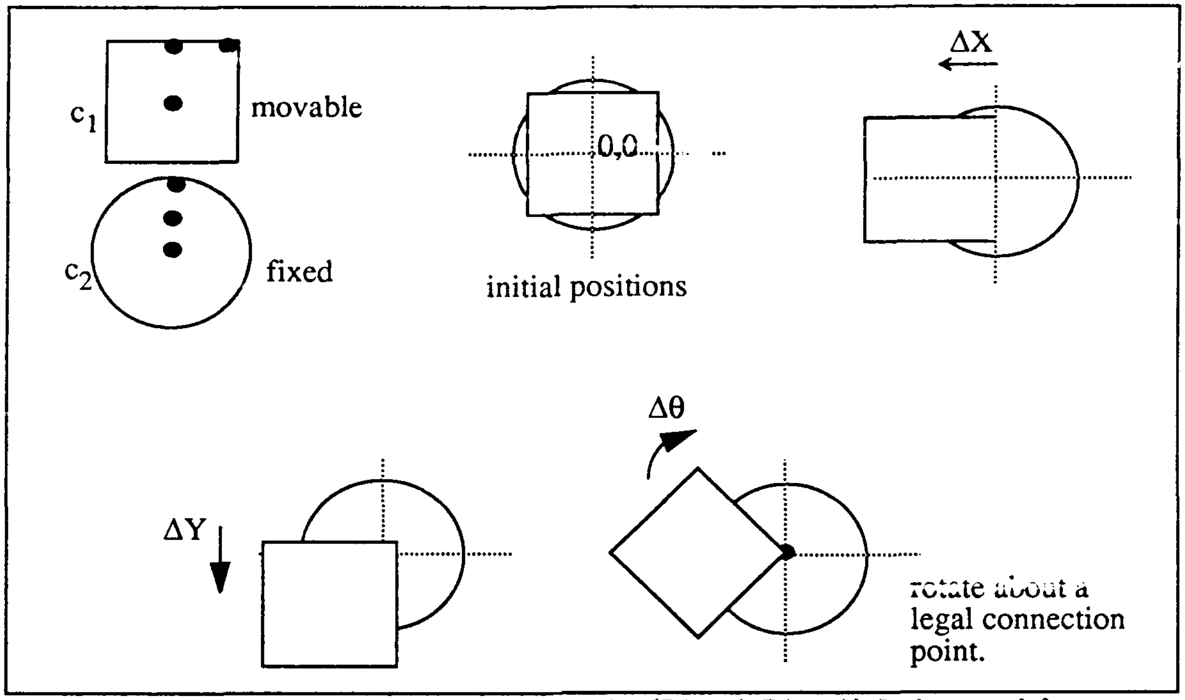

Figure 2a: Example showing how to define a *construction* $\{R(c_1,c_2), R(c_2,c_3)\}$. In the upper left, two geons and their legal attachment points are shown. Two *components* are assembled by $R(c_1,c_2) = (\Delta x, \Delta y, \Delta\theta)$. The two translation operations bring connection points into correspondence. The moveable component is then oriented by a rotation about this point.

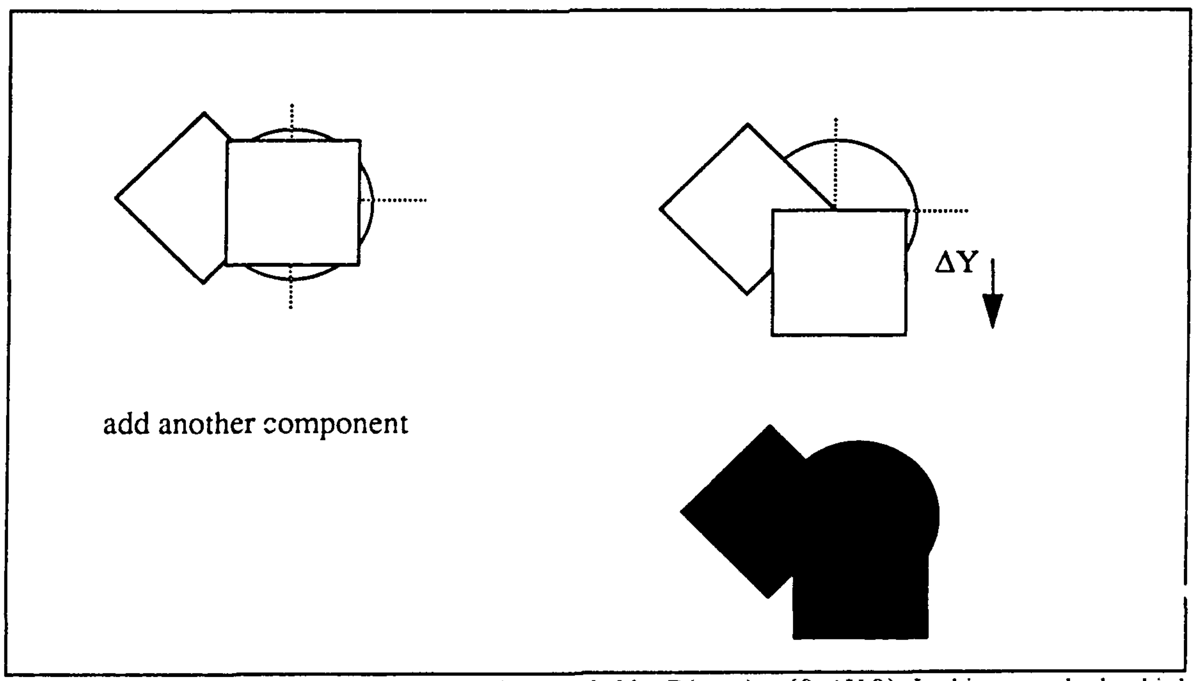

Figure 2b: The two-*component construction* is extended by $R(c_2,c_3) = \{0, \Delta Y, 0\}$. In this example the third *component* is added to the previously moved one (square) with ΔY as the only displacement. Centroids are used to define initial positions in both figures. A final silhouette shape is yeilded by a filling operation.

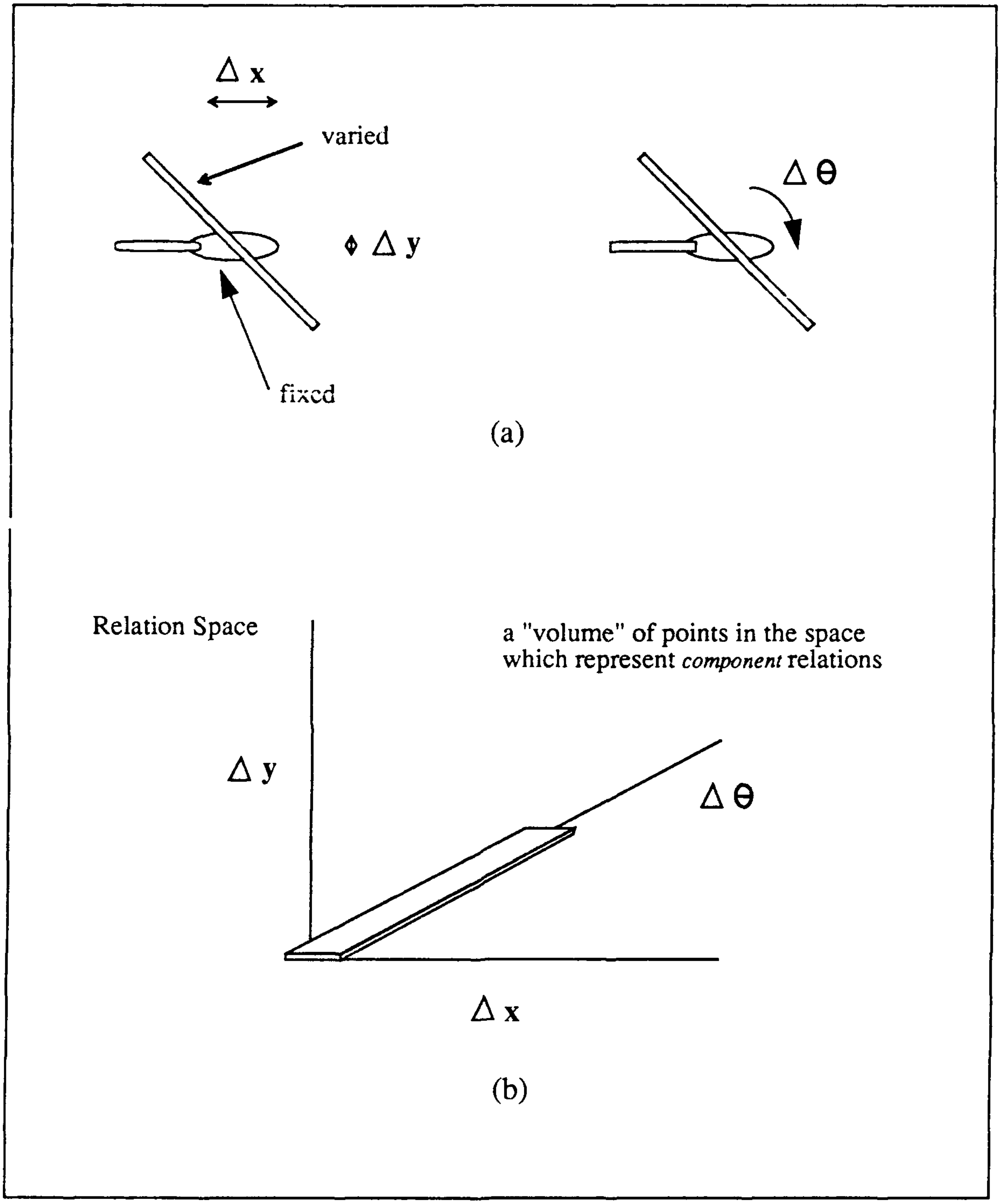

Figure 3: This example illustrates articulation of *components* in a simple model of a "helicopter" viewed from the top. (a) The "rotor" *component* is permitted a small variation in displacement with respect to the defined "x" axis of the fuselage, but virtually no displacement in the "y" axis. However, the rotor is permitted all angular displacements about its centroid. (b) These *relations* may be described by a "volume" in *relation* space; in this case by a thin wafer. To accommodate the tail component, the *relation* space could be expanded by another three dimensions to show its *relations* with respect to the fuselage.

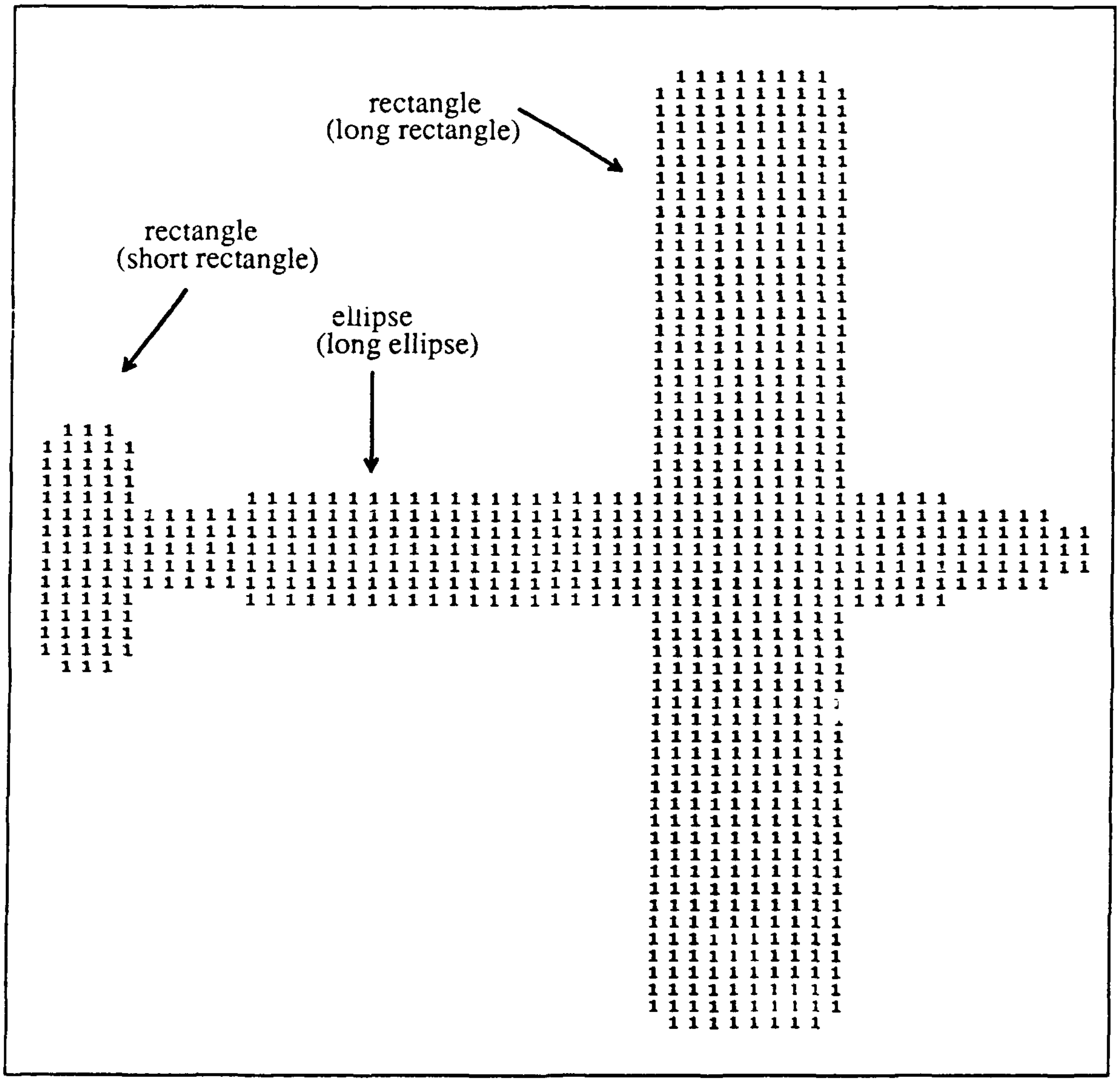

Figure 4: An example of an input image (low-resolution) for RBC-R to recognize. The geon *types* are indicated with their *instantiations* in parentheses.

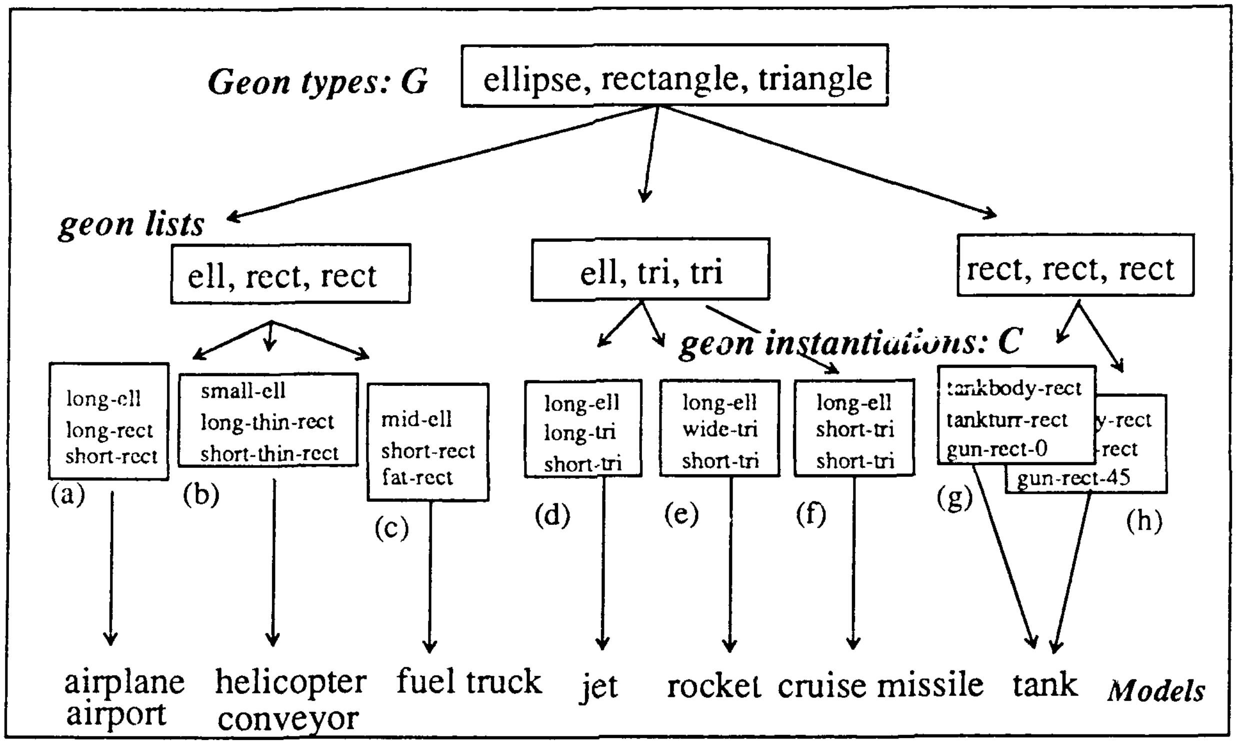

Figure 5: Domain hierarchy for the implementation. The geon *types* comprise the set of high resolution geons that a (hypothetical) *component* recognition system might discriminate. The geon lists are groupings of three geons. Geon *instantiations* are particular instances of the geons from a list. These are used to create the *construction* universe and define the models of the domain. The letters refer to distinct *construction* sets.

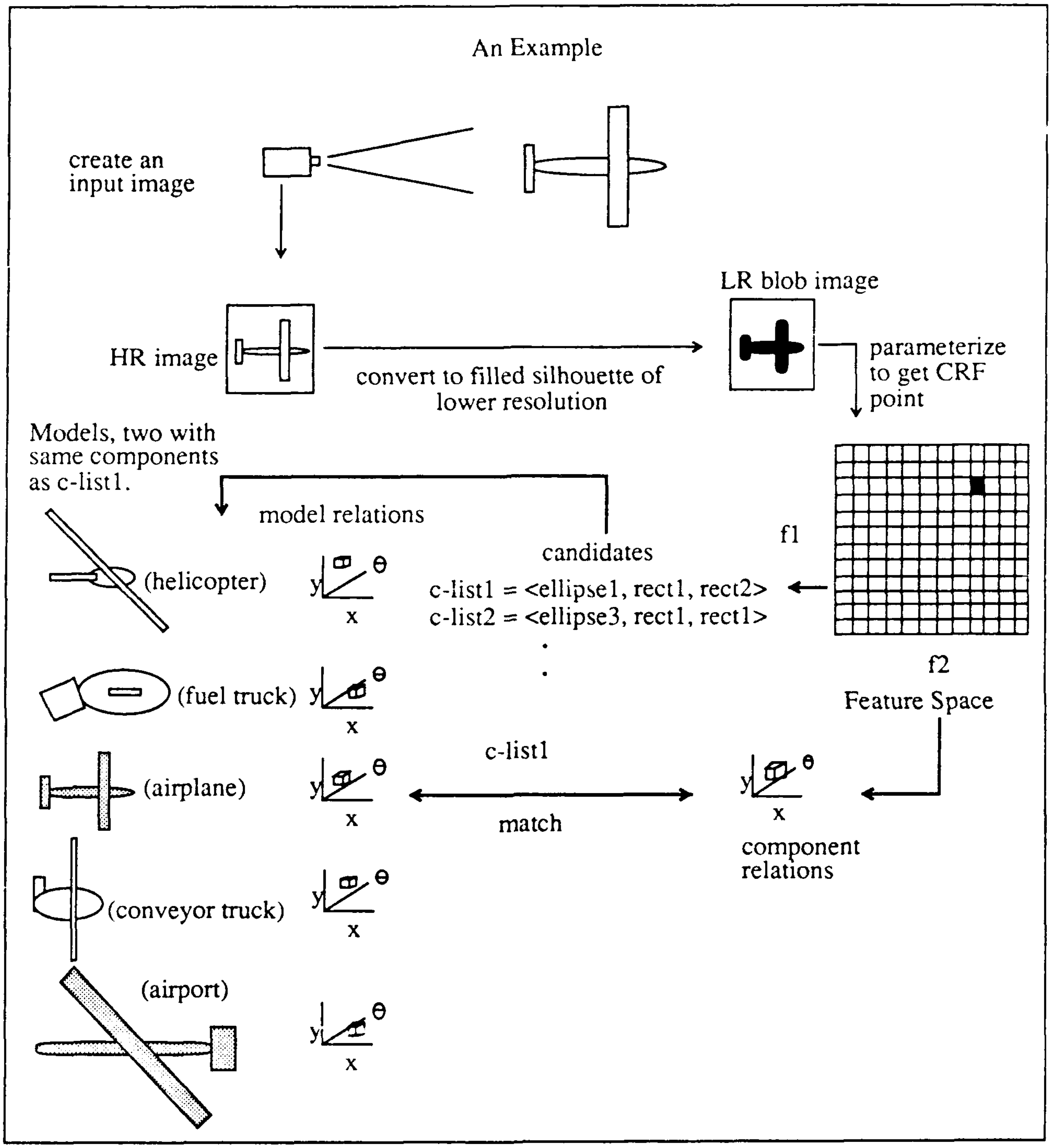

Figure 6: Shape features evoke-lists in a feature space. These are compared against model c-lists. Shaded models on the left are found to have c-list1. No models have c-list2. For c-list1, the *component relations* are compared between the candidates and the two models. Only one match is confirmed: the "airplane".

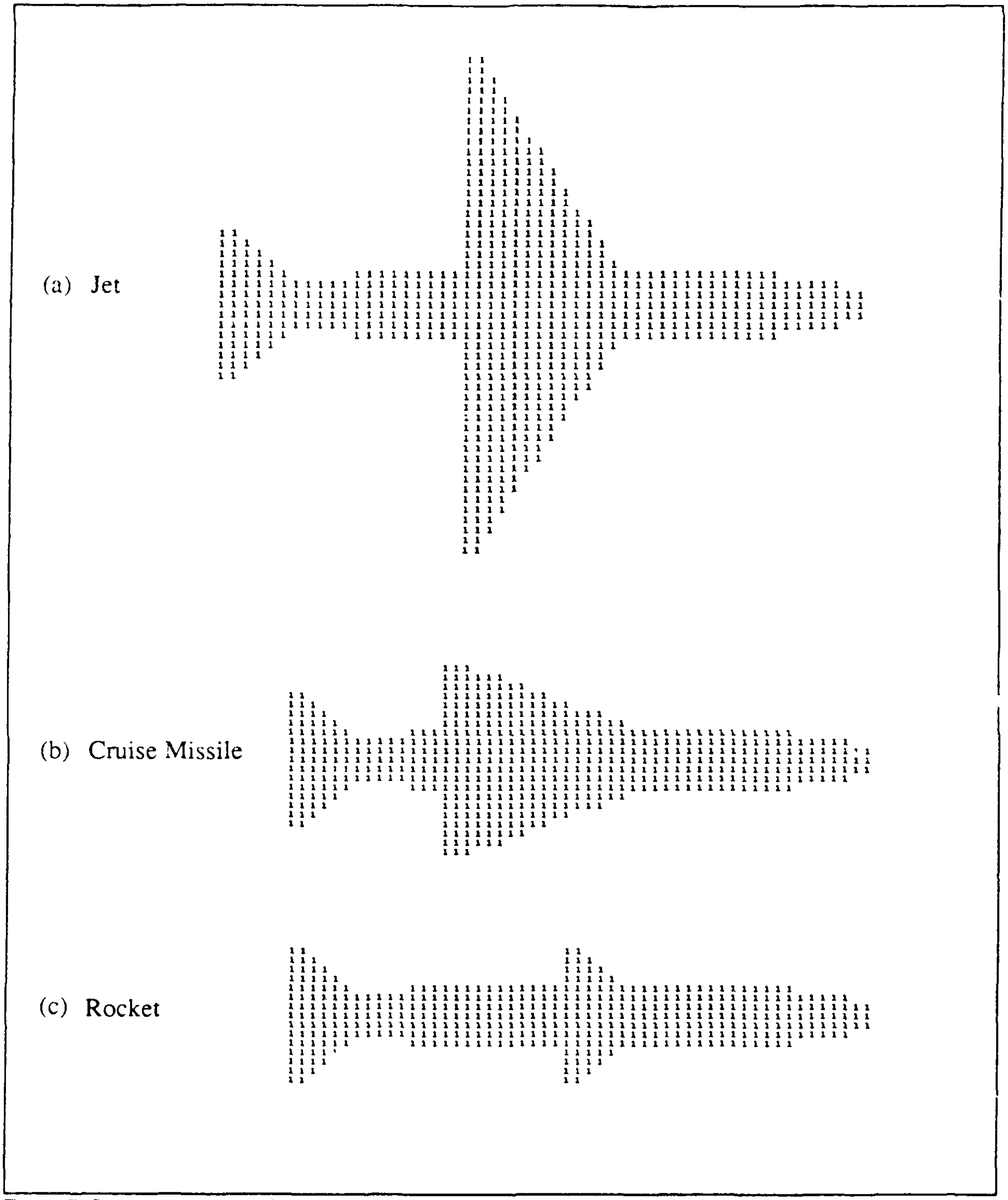

Figure 7: Some test images submitted for recognition. For these images the same *construction* operator was used as for creating the *construction* universe. In other words these images were among the K_i set used to establish the CRF *map*. Their correct identification illustrates that RBC-R can reliably recognize K_i that it was "trained" on.

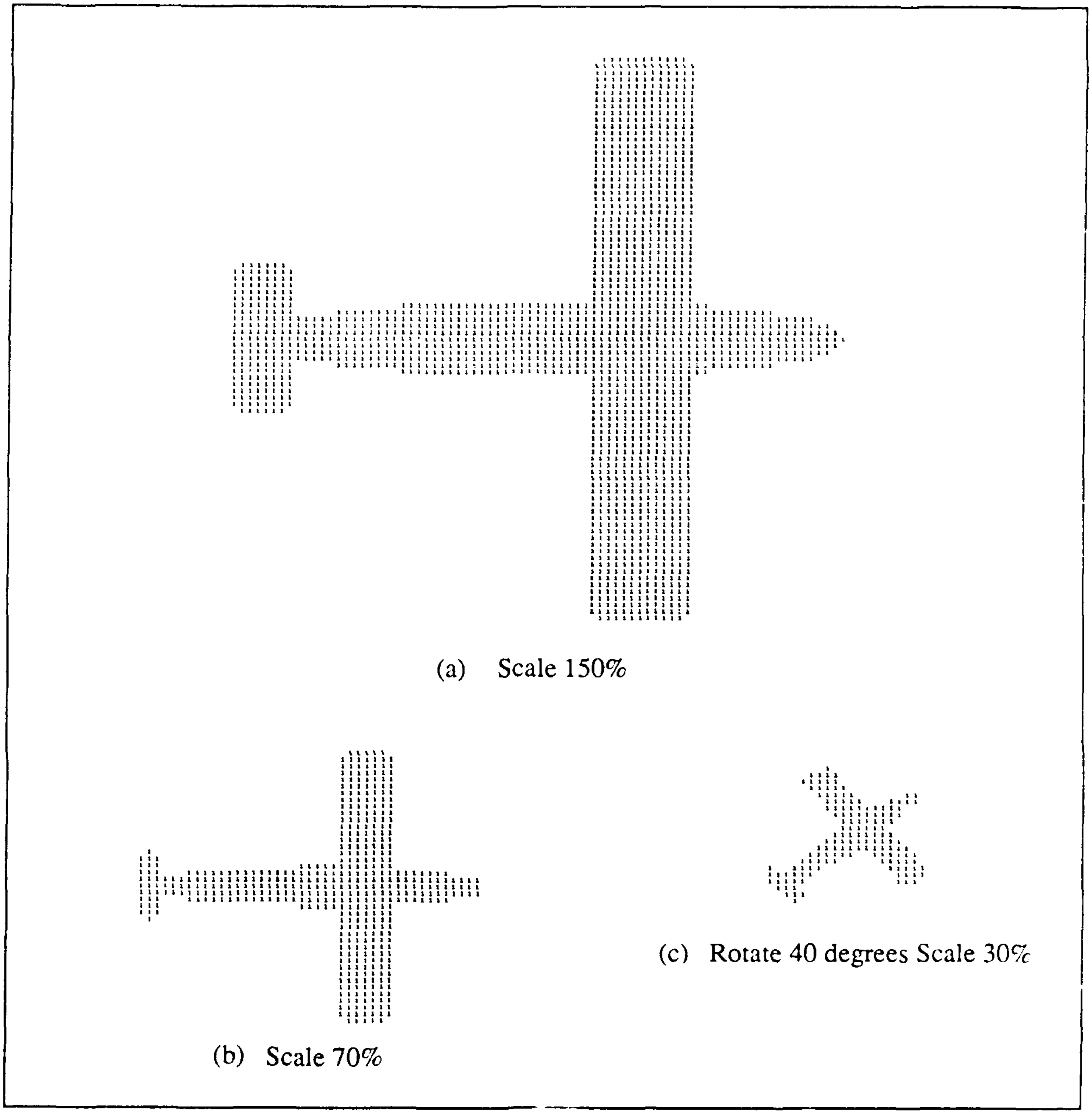

Figure 8: Example input images at different scales and rotation. Scaling is expressed as a percentage of the size of the K_i used to create the CRF *map*. None of these images was used in establishing the CRF *map*, but are each novel input images. Images (a) and (b) resulted in shape features with identical coordinates to the "airplane" K_i in the CRF *map* and recognition was confirmed by comparing this against model knowledge. Image (c) resulted in feature coordinates in the neigborhood of the "airplane" K_i and after a local search of the CRF *map* RBC-R found the appropriate K_i and confirmed the match against model knowledge.

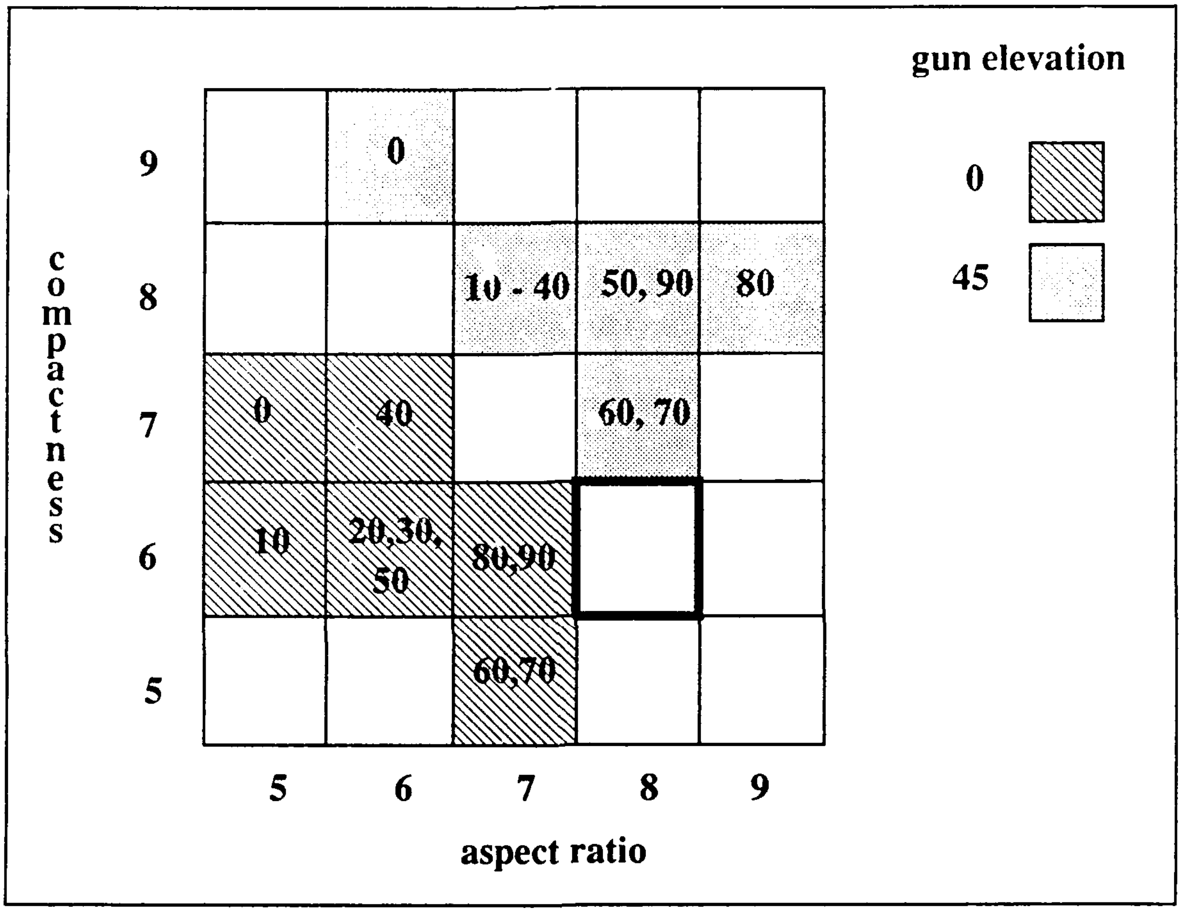

Figure 9: This is a summary of the mapping of shapes into the CRF *map* for all "tank" K_i used to define the "tank" model. The two c-lists, one for each "gun elevation", are differentiated by shading. The angle between the "tank body" and "turret" are given numerically. It can be seen that the parameters vary smoothly, albeit non-linearly. The choice of the two "gun" angles happens to be nearly optimal for this quantization of the CRF *map* as a clean but neighboring border separates the two c-list image sets. It can also be seen that as the angle increases, aspect ratio tends to increase and compactness shows variance but no simple linear change. A novel "tank" image with "turret" angle 70 degrees and intermediate "gun" elevation of 25 degrees is indicated by the bold box.

Appendix 1: Definitions

To avoid potential confusion, terms which have a specific technical meaning in the text are given in italics and defined below. These should be distinguished from general dictionary usage. All technical terms are also defined in the text but are given here for reference.

component - In general usage, this is an elementary constituent of an *object*. As defined for RBC-R, this term means a unit or part of a *construction* as defined by an *instantiated* geon *type* (sect 1.1)

construction - This is a special term defined for RBC-R which refers to any arbitrary assembly of connected *components* (sect 2.2).

instantiation - As used for RBC-R this refers to a specific instance of a geon *type* to which a metric has been applied (sect 1.2).

interpretation - For RBC-R this term refers to the process of determining the identity of an *object*. It is synonymous with recognition. It amounts to finding a model for an unknown input image (sect 1.1).

map - This refers to the *construction relation* feature map, the CRF map (sect 2.4.2).

object - An entity in the domain for which a description is possible in terms of a configuration of *components* and therefore which may have a model defined in terms of these *components* and their *relations* (sect 1.1).

relation - The spatial, two dimensional position and orientation of one *object component* relative to another. It is defined by an affine transformation of translation and/or rotation with respect to *component* centroids.

resolution - This term is used as an adjective to describe the silhouette image of a single geon or *object*. This image property is implied, therefore, in cases where "low resolution geon" is mentioned. Lower resolution images contain less information in terms of features which could be used to distinguish between different images.

type - This refers specifically to geons as defined by Biederman (1987). The term is used in contradistinction to *instantiations* in RBC-R (sect 1.2).

Appendix 2: Example Recognition

The input image was created using the program drawing functions to produce an image like that shown in fig. 4. The silhouette in this image was then rotated 40 degrees and scaled to 30 percent. The following is a record from a printout of RBC-R during the *interpretation* process.

Searching feature space at features: 3,9
offset : (0 0)

Candidate component lists found in CRF map : none
offset : (-1 0)

Candidate component lists found in CRF map :
(LONG-ELL LONG-RECT SHORT-RECT)
(LONG-ELL LONG-TRI SHORT-TRI)

trying to match components elicited by input to models

Model with components matched : AIR-PORT
Model with components matched : AIRPLANE

Check for relation matches

>>> MATCH FOUND <<<

Name : AIRPLANE
Components :
LONG-ELL (geon 1) off-center connects to LONG-RECT (geon 2) middle 0 degrees
LONG-ELL (geon 1) end connects to SHORT-RECT (geon 3) middle 0 degrees

----

trying to match components elicited by input to models

Model with components matched : FIGHTER
Check for relation matches

>>> MATCH FOUND <<<

Name : FIGHTER
Components :
LONG-ELL (geon 1) middle connects to LONG-TRI (geon 2) middle 90 degrees
LONG-ELL (geon 1) end connects to SHORT-TRI (geon 3) middle 90 degrees

----

offset : (-1 1)
Candidate component lists found in CRF map :
(LONG-ELL LONG-RECT SHORT-RECT)

trying to match components elicited by input to models

Model with components matched : AIR-PORT
Model with components matched : AIRPLANE

Check for relation matches

>>> NO MATCH <<<

offset : (0 1)

Candidate component lists found in CRF map : none
offset : (1 1)

Candidate component lists found in CRF map : none
offset : (1 0)

Candidate component lists found in CRF map : none
offset : (1 -1)

Candidate component lists found in CRF map : none
offset : (0 -1)

Candidate component lists found in CRF map : none
offset : (-1 -1)

Candidate component lists found in CRF map :
(LONG-ELL LONG-RECT SHORT-RECT)

trying to match components elicited by input to models

Model with components matched : AIR-PORT
Model with components matched : AIRPLANE

Check for relation matches

>>> NO MATCH <<<

Thus, the airplane and jet models only are resolved as interpretations of the input image. This required searching the feature space coordinates adjacent to the initial coordinates which failed to locate any K_i for a possible match to models. All eight surrounding coordinates are examined with an arbitrary starting point. At coordinates (2,9) K_i are found which have models. No other coordinates were found to produce a model match on the basis of both *components* and *relations*.

References

Ballard, D.H & Brown, C.M. (1982) **Computer Vision.** Prentice-Hall: Englewood, NJ.

Bhanu, B. (1984) Representation and shape matching of 3-D objects. IEEE **Transactions on Pattern Analysis and Machine Intelligence. 6** 340-351.

Bhanu, B. & Faugeras, O.D. (1984) Shape matching of two dimensional objects. **IEEE Transactions on Pattern Analysis and Machine Intelligence. 6**, 137-156.

Biederman, I. (1985) Human image understanding: Recent research and a theory. **Computer Vision, Graphics and Image Processing. 32**, 29-73.

Biederman, I. (1987) Recognition-by-components: A theory of human image interpretation. **Psychological Review. 94** 115-148.

Brooks, R.A. (1981) Symbolic reasoning among 3-D models and 2-D images. **Artificial Intelligence, 17** 285-348.

Brooks, R.A. (1984) **Model-Based Computer Vision.** UMI Press: Ann Arbor MI.

Browse, R.A. (1982) Knowledge-based visual interpretation using declarative schemata. PhD thesis, **Technical Report 82-12.** Depart Computer Science, Univ British Columbia.

Browse, R.A. & Cutmore, T.R.H. (paper in preparation)

Browse, R.A. & Rodrigues, M.G. (1987) Propagation of interpretations based on graded resolution input. **Proceedings of the First International Conference on Computer Vision.**

Cutmore, T.R.H (1989) Using low resolution images to constrain spatial relations in visual interpretation. **M.Sc. thesis**, Dept Computer Science, Queen's Univ, Kingston.

Goshtasby, A. (1985) Description and discrimination of planar shapes using shape matrices. **IEEE Transactions on Pattern Analysis and Machine Intelligence. 7** 738-743.

Horn, B.K.P. (1986) **Robot Vision.** The MIT Press: Cambridge, MA.

Kanade, T. (1977) Model representations and control structures in image understanding. **Proceedings of the International Joint Conference on Artificial Intelligence. 5** 1074-1082

Kelly, M.D. (1971) Edge detection in pictures by computers using planning. In R. Meltzer and D. Michie (Eds.) **Machine Intelligence 6,** New York: Elsevier, 379-409.

Kron, G. (1939) **Tensor Analysis of Networks.** John Wiley & Sons, Inc: New York.

Ma, J., Wu, C. & Lu, X. (1986) A fast shape descriptor. **Computer Vision and Image Processing. 34**, 282-291.

Marr, D., & Nishihara, H.K. (1978) Representation and recognition of the spatial organization of three dimensional structure. **Proceedings of the Royal Society of London (Series B) 200**, 269-294.

Neveu, C.F., Dyer, C.R. & Chin, R.T. (1986) Two-dimensional object recognition using multiresolution models. **Computer Vision, Graphics and Image Processing. 34**, 52-65.

Roberts, L.G. (1965) Machine perception of three dimensional solids. In: J.T. Tippet et al. (Eds.) **Optical and Electro-Optical Information Processing**, MIT Press, Cambridge. 159-197.

Smith, I.G.R., and Browse, R.A. (1988) Invocation matching using geon structure. **Technical note 88-222.** Department of Computing and Information Science. Queen's University, Kingston, Ontario.

Steele, G.L. (1984) **Common Lisp: The Language.** Digital Press. Bedford MA.

Tanimoto, S.L. & Pavlidis, T. (1975) A hierarchical data structure for picture processing. **Computer Graphics and Image Processing. 2**, 104-119.

Tsai, W. & Yu, S. (1985) Attributed string matching with merging for shape recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence. 7** 453-462.

Wang, Y.F., Magee, M.J. & Aggarwal, J.K. (1984) Matching three-dimensional objects using silhouettes. **IEEE Transactions on Pattern Analysis and Machine Intelligence. 6**, 513-518.

Acknowledgements

This research was part of a Master's thesis in Computer Science at Queen's University, Kinston Ontario. I wish to thank my thesis supervisor Dr. Roger Browse and D.C.I.E.M. for financial support during this period.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 12.)

   DCIEM, 1133 Sheppard Ave. W., North York, Ont. M3M 3B9 CANADA

2. DOCUMENT SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable)

   Unclas.

3. DOCUMENT TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title.)

   Shape recognition by computer in simulated aerial images

4. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

   Research Report

5. AUTHOR(S) (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

   Cutmore, Tim R.H.

6. DOCUMENT DATE (month and year of publication of document)

7a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

   29

7b. NO. OF REFS (total cited in document)

   25

8a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

   Proj # 51CJ 13

8b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

9a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

   DCIEM No. 91-05

9b. OTHER DOCUMENT NO.(S) (Any other numbers which may be assigned this document either by the originator or by the sponsor)

10. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

   (X) Unlimited distribution
   ( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
   ( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
   ( ) Distribution limited to government departments and agencies; further distribution only as approved
   ( ) Distribution limited to defence departments; further distribution only as approved
   ( ) Other

11. ANNOUNCEMENT AVAILABILITY (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availabilty (10). However, where further distribution (beyond the audience specified in 10) is possible, a wider announcement audience may be selected.)

   Unlimited

12. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

   Same as 1.

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U) It is not necessary to include here abstracts in both offical languages unless the text is bilingual).

Visual recognition of objects by a machine involves classifying an input using knowledge about the kinds of objects expected in the domain. Model-based systems maintain a knowledge of objects in the domain in the form of a representation which can be compared to the unknown input. Since a given object type may appear in a variety of forms and under a variety of viewing conditions some efficient yet flexible means of guiding the recognition process to consider and then verify the object identity is necessary.

The utility of low-resolution shape information to constrain object recognition was investigated in the context of a system which is predicated upon a component description of objects. A computationally intensive prepass using a syntax for combining components yields a universe of "constructions" which are coded into a construction relation feature (CRF) map. Each construction is coded into the N-dimensional map according to a shape parameterization of its low-resolution image (each dimension codes a shape feature). From a subset of these constructions the object models are specified in terms of their component structure. The CRF map thus links the low resolution shapes of instances of an object to its object model.

To recognize an unknown object the input is first converted to low resolution. Then, shape parameters are taken (for example, in terms of its relative elongation and compactness). The CRF map is examined in the region of these feature coordinates for possible matches with object models, first on the basis of a component match followed by a verification of the relations between components. An application to identifying objects in simulated aerial photographs of military airports is presented.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible, keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Computer Vision Shape Resolution Image Model-Based